# Aufgaben logical indexing

## Primzahlen

Erzeugen Sie mit dem Kommando Primes=primes(1000) einen Vektor Primes, der alle Primzahlen kleiner 1000 enthält.

Die Zahlen innerhalb von Primes sind der Größe nach geordnet.

a)

Erzeugen Sie mittels logical indexing aus dem Vektor Primes einen Vektor PrimesGreater170, der alle Primzahlen enthält, die größer als 170 sind!

b)

Erzeugen Sie mittels logical indexing aus dem Vektor Primes einen Vektor PrimesLess250, der alle Primzahlen enthält, die kleiner als 250 sind!

c)

Erzeugen Sie mittels logical indexing aus dem Vektor Primes einen Vektor Primes150to250, der alle Primzahlen zwischen einschließlich 150 und 250 enthält! (Zusatzfrage: warum ist das „einschließlich“ Quatsch?)  
Hinweis: Die Bedingung „zwischen“ besteht aus zwei Einzelbedingungen. Diese müssen logisch verundet werden, dazu dient der &-Operator.

Beispiel &-Operator:

index1=[1,0,0,01,1,1,0];

index2=[0,0,0,1,1,1,1];

index1 & index2

ans =

1×7 logical array

0 0 0 1 1 1 0

d)

Lesen Sie die Matlabdokumentation zur Funktion „ismember“. Nutzen Sie diese Funktion, um den Vektor NonPrimes zu erstellen, der alle Zahlen bis 1000 enthält, die keine Primzahl sind!

Hinweis: Der ~ Operator negiert ein logical array

Beispiel:  
>> ~logical([1,1,1,0,0,0])  
ans =  
 1×6 logical array  
 0 0 0 1 1 1

e)

Ob eine Zahl durch 2,3,5,oder 11 teilbar ist, kann man einfach mittels der Teilbarkeitsregeln feststellen. In Matlab geht das schneller mit der mod-funktion. mod(z1,z2) gibt den Teilerrest beim Berechnen von Z1 durch Z2 an. So ergibt mod(10,3) z. B. 1. Ist z2 ein Teiler von z1, so ist mod(z1,z2)==0.

Erstellen Sie einen Vektor AlmostPrime mit Zahlen zwischen 1 und 1000, die auf den ersten Blick als Primzahl durchgehen, weil sie nicht durch 2,3,4 oder 11 teilbar sind, aber in Wirklichkeit keine Primzahlen sind!

## Beschneiden von Messkurven

Gegeben ist ein Programm zum Einlesen und Darstellen von realen Messdaten. Bitte lesen Sie damit die Datei Messdaten.mat ein! (Sie müssen das Programm nur starten und in der sich öffnenden Maske zu der Datei Messdaten.mat navigieren, die Sie vorher aus Studip übernommen haben.)

%% load Data

clear all;

clc;

[FileName,PathName] = uigetfile('\*.mat','Suche Deine Datei aus');

load ([PathName,FileName]);

t=out.ScopeData1.time;

Y=(out.ScopeData1.signals(1).values);

X=out.ScopeData1.signals(2).values;

plot(t,Y,t,X)

%% cut data

tstart=34.0; %nur Beispielwert

tend=195; %nur Beispielwert

%Beschneiden Sie nun die Vektoren t, X, Y so, dass nur noch die Werte

%zwischen tstart und tend benutzt werden.

plot(t,Y,t,X)

Es wird dann das folgende Diagramm erzeugt.



Rechnen Sie nun t, X und Y so um, dass alle Messwerte vor tstart und nach tend verworfen werden. Das heißt das zweite plot-Kommando soll dann das folgende Diagramm erzeugen:



## Beschädigte Messdaten

Das folgende Codesegment liest t, X und Y Daten ein. (Wählen Sie die Datei Messdatenbeschädigt.mat aus).

%% load Data

clear all;

clc;

[FileName,PathName] = uigetfile('\*.mat','Suche Deine Datei aus');

load ([PathName,FileName]);

Die Daten enthalten NaN Einträge. Sehen Sie sich in der Matlabdokumentation die Funktion isnan an!

Beseitigen Sie jede Einzelmessung, bei der t oder X oder Y keine Zahl ist. (Hinweis, Sie müssen drei Bedingungen verunden.)